



Tabla de Contenidos

- Introducción
- Algoritmos de detección de esquinas
- SIFT



Introducción

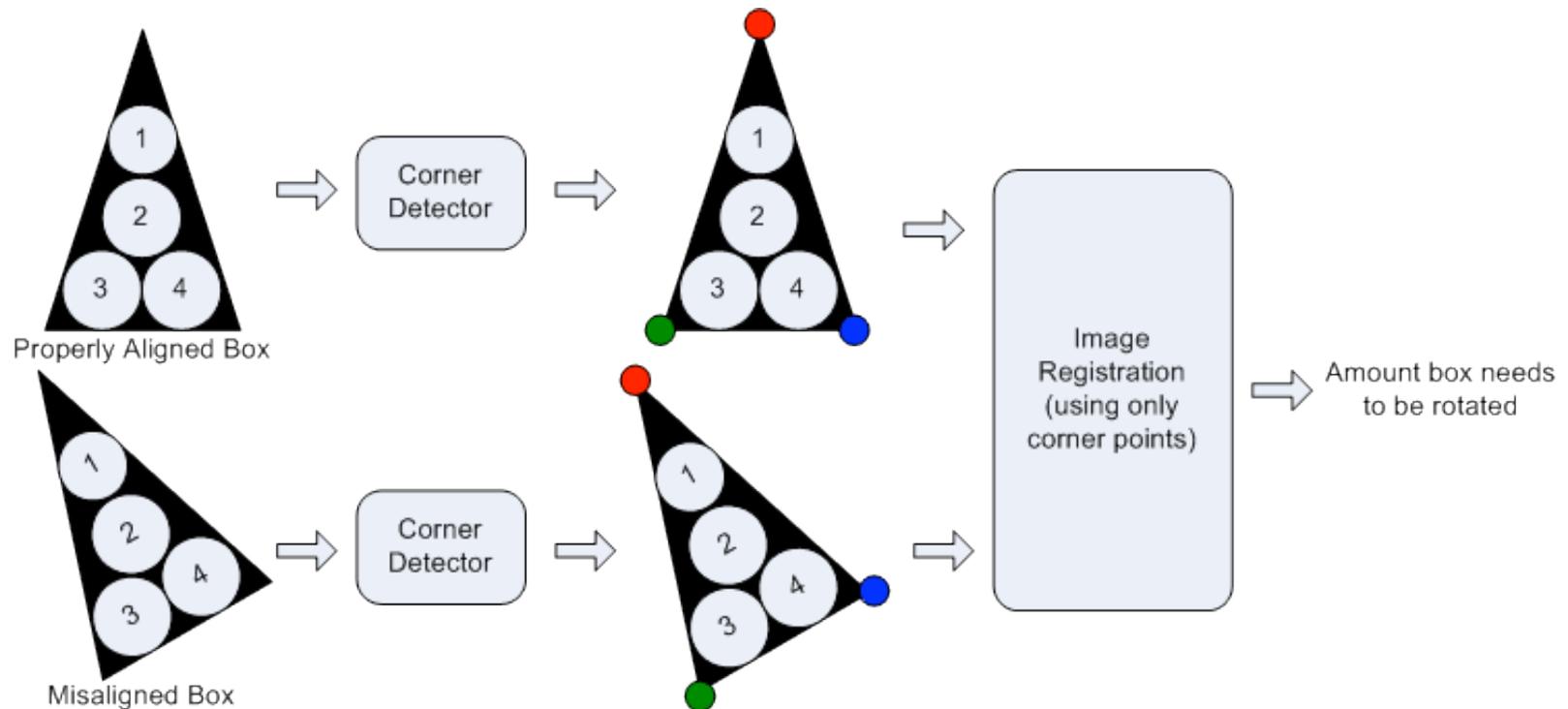
- Las esquinas como puntos de interes
 - Correspondencia entre dos imágenes
 - Punto a punto: prohibitivo en la mayoría de los casos
 - Alternativa: sólo emparejar las zonas más relevantes
 - Método: utilizar *puntos de interés*
 - Cómo detectarlos?
 - Entre otras alternativas, se usan mucho las esquinas
 - Aplicaciones
 - Correspondencia estéreo
 - Registro de imágenes
 - Generación de mosaicos o panoramas
 - Tracking
 - Navegación
 - ...





Introducción

○ Caso de uso





Introducción

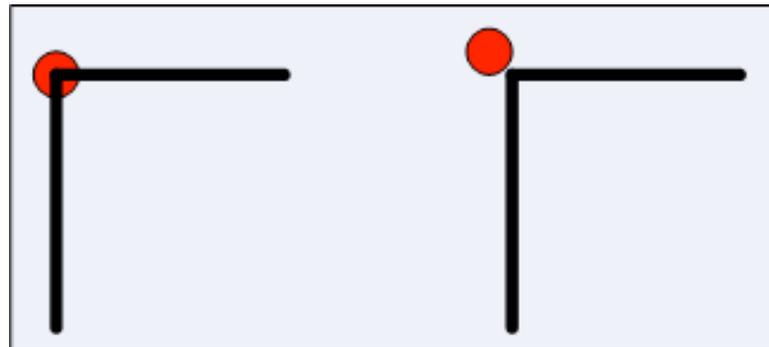
- Requisitos de un detector de esquinas
 - Detección de todas las esquinas visibles
 - No aparición de falsos positivos
 - Correcta localización de todas las esquinas
 - Repetibilidad
 - Detección robusta frente al ruido
 - Eficiencia computacional



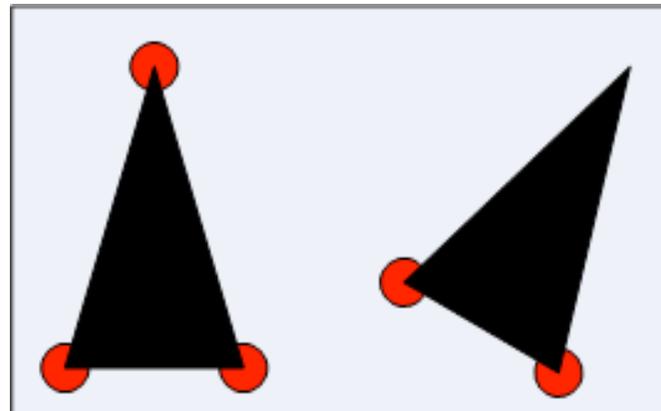


Introducción

- Casos
 - Localización correcta o incorrecta



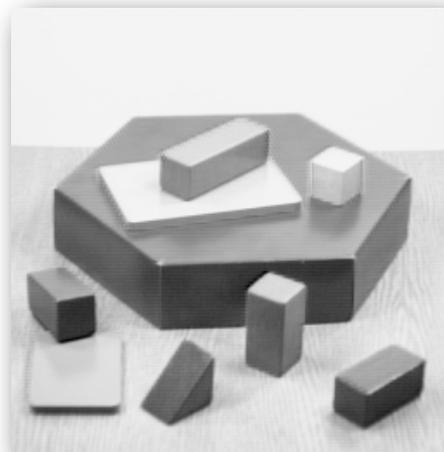
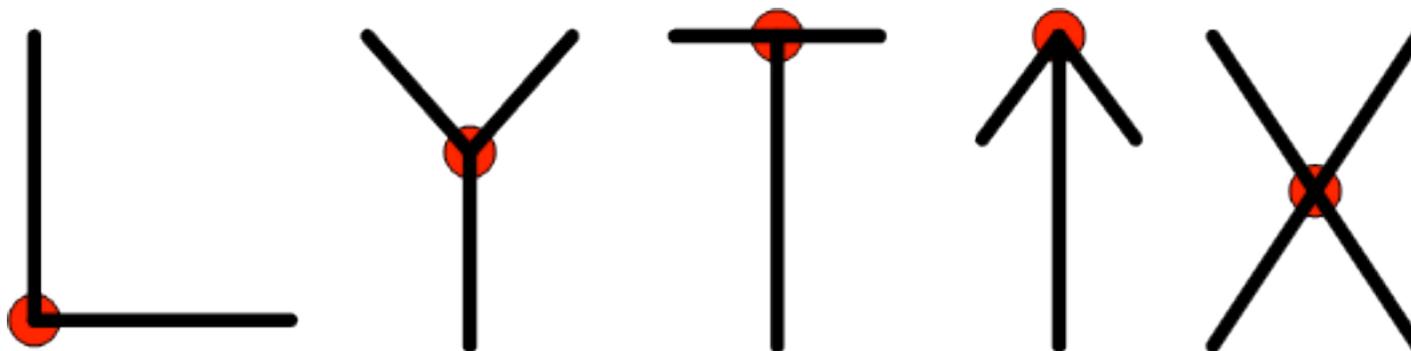
- Repetibilidad





Introducción

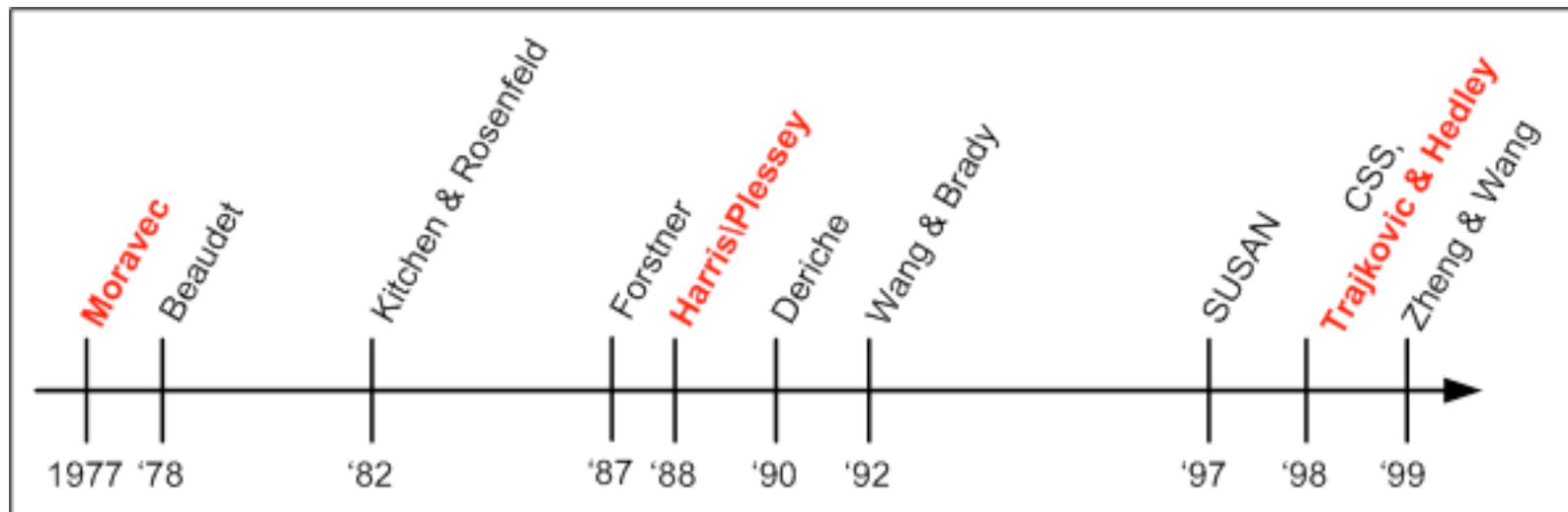
- Evaluación de un detector de esquinas
 - Comportamiento frente a distintos tipos de esquinas





Introducción

○ Evolución



○ Aproximaciones fundamentales

- Métodos de borde-relación
- Métodos topológicos
- Métodos de autocorrelación





Introducción

- Métodos de borde-relación
 - Kitchen & Rosenfeld
 - Basado en el cambio de dirección del gradiente
 - ⊙ Para puntos de un contorno
 - ⊙ Ponderado con su magnitud en el punto
 - ⊙ El máximo de esta medida marca una esquina
 - ⊙ Baja repetibilidad y localización pobre
 - Wang & Brady
 - Basado en la curvatura
 - ⊙ Apto para tiempo real
 - ⊙ Mejora la calidad del K&R





Introducción

- Métodos topológicos
 - Beaudet
 - Sienta las bases para detectores posteriores
 - Invariante a rotación, determinante del Hessiano
 - Sensible al ruido y a la escala
 - Deriche
 - Extensión de Beaudet, por aplicación a varias escalas
 - Mejora la detección respecto a Beaudet
 - Sigue siendo sensible al ruido





Introducción

- Métodos de autocorrelación (I)
 - Moravec
 - Introduce el concepto de punto de interés
 - Mide cambios de intensidad al desplazar una ventana
 - ⊙ Autocorrelación en las 8 direcciones
 - Se asigna a cada pixel el valor mínimo de estos cambios
 - Los puntos de interés son los máximos locales
 - Sensible al ruido
 - Harris & Plessey
 - Calcula la autocorrelación mediante las derivadas
 - ⊙ Por desarrollo de Taylor del cálculo de Moravec
 - Se le considera el mejor detector de esquinas reales
 - Mala localización y alto coste computacional





Introducción

- Métodos de autocorrelación (y II)
 - Zheng & Wang
 - Medida simplificada del operador de Harris
 - Empeora la detección
 - Mejora la eficiencia y la localización
 - Forstner
 - Usa una medida similar a la de Harris
 - Calcula el umbral localmente mediante estadísticos
 - Mejor localización a mayor coste computacional





Introducción

- Métodos alternativos (I)
 - CSS (*Curvature Scale Space*)
 - Esquinas=máximo local de la curvatura absoluta
 - ⊙ Una esquina es un cambio brusco de dirección de un borde
 - Pasos a seguir
 - ⊙ Extraer contornos (p.e. Canny)
 - ⊙ Calcular la curvatura a la mayor escala
 - ⊙ Esquinas = máximos locales por encima de un umbral
 - ⊙ Confirmar esquinas a distintas escalas (mejor localización)





Introducción

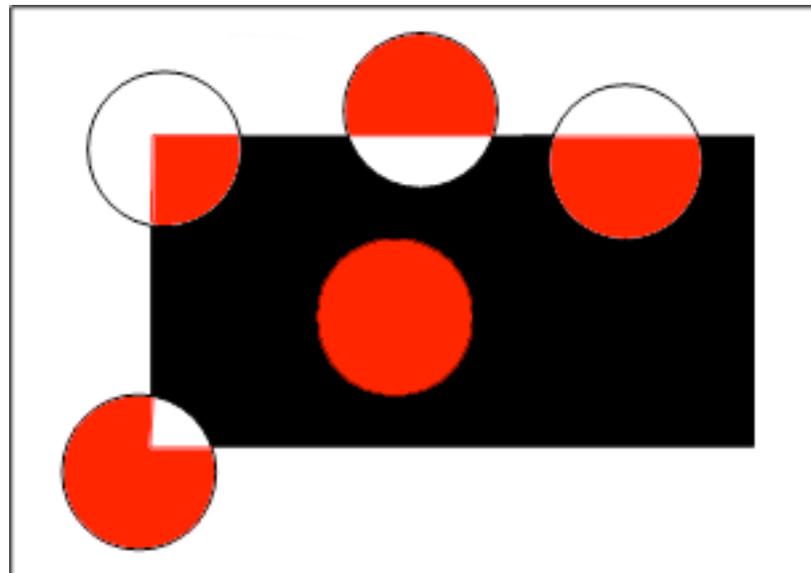
- Métodos alternativos (II)
 - Smith & Brady (SUSAN)
 - Compara el brillo en máscaras circulares sobre cada pixel
 - Dentro, los píxeles del mismo objeto tienen mismo brillo
 - Se calcula cuántos tienen mismo brillo que el central
 - A este valor se le llama USAN (*Univalue Segment Assimilating Nucleus*)
 - Esquina= mínimo local en el valor de USAN
 - SUSAN= *Smallest USAN*
 - Se usa también para detectar bordes
 - Es robusto al ruido pero con repetibilidad moderada





Introducción

- Métodos alternativos (III)
 - Uso de SUSAN





Introducción

- Métodos alternativos (y IV)
 - Trajkovic & Hedley
 - Similar a SUSAN
 - Variaciones de brillo en las líneas que pasan por un pixel
 - En una esquina las variaciones en cada línea serán grandes
 - Menos robusto que Harris
 - Uno de los detectores más rápidos





Introducción

○ Comparación de los detectores

Operador	Detec.	Localización	Repet.	Robust.	Speed
Beaudet	+	+	- escalado, ++ afín	-	++
Moravec	+	++	+	+	++
K&R	+	+	+	+	-
Forstner	++	++	+++ afín, + escalado	++	-
Plessey	++	++ (sólo esquinas L)	+++ afín, + escalado	+	-
Deriche	+	++	++	-	++
W&B	++	++	++	+	++
SUSAN	++	++ (-imágenes borrosas)	++ escalado, - afín	+++	++
CSS	++	++	+++	++	?
T&H v4	-	++	+ (no rotación)	-	+++
T&H v8	+	++	+ (no rotación)	++	+++
Z&W	++	++ (sólo esquinas L)	+++ afín, + escalado	+	+





Tabla de Contenidos

- Introducción
- Algoritmos de detección de esquinas
- SIFT





Algoritmos de detección de esquinas

- En detalle
 - Por motivos históricos, aceptación o rendimiento
 - Asignan valor a cada pixel en la imagen
 - No se basan por tanto en análisis de curvatura del borde
 - Se van a estudiar
 - Moravec (1977)
 - Harris/Plessey (1988)





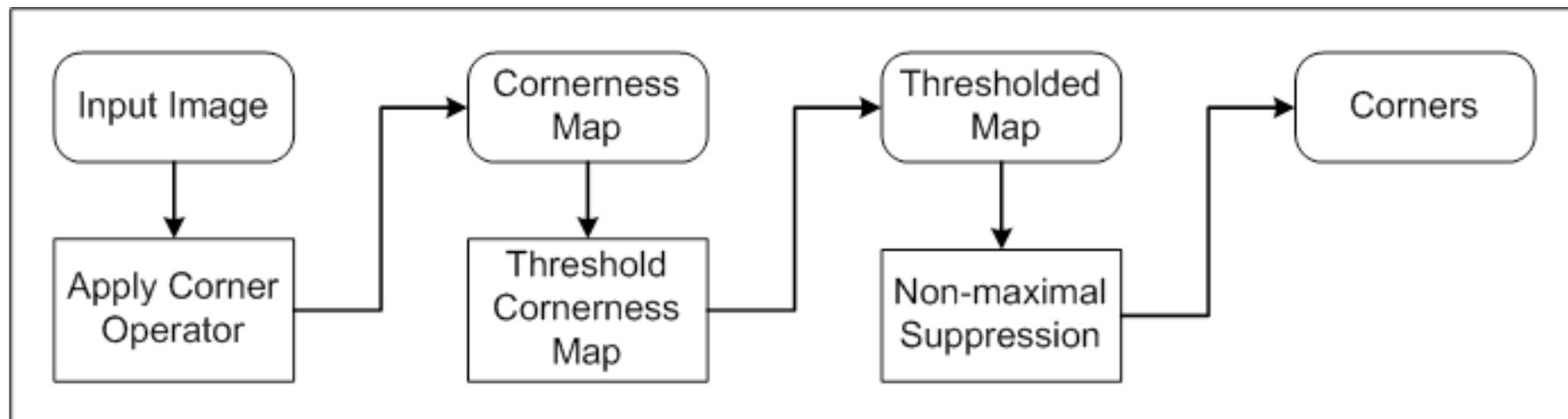
Algoritmos de detección de esquinas

- Pasos en la detección (todos)
 1. Aplicar el operador de esquinas
 - Tomar la imagen y definir valores que usa el algoritmo
 - Aplicar a cada pixel en la imagen: *cornerness map*
 2. Umbralizar el *cornerness map*
 - Las esquinas son sus máximos locales
 - Se incluyen algunos débiles, que no son esquinas reales
 - Se umbraliza para eliminarlos
 3. Supresión de no máximos
 - Existen regiones que superan el umbral
 - Para definir un puntos aislados se quitan los no máximos



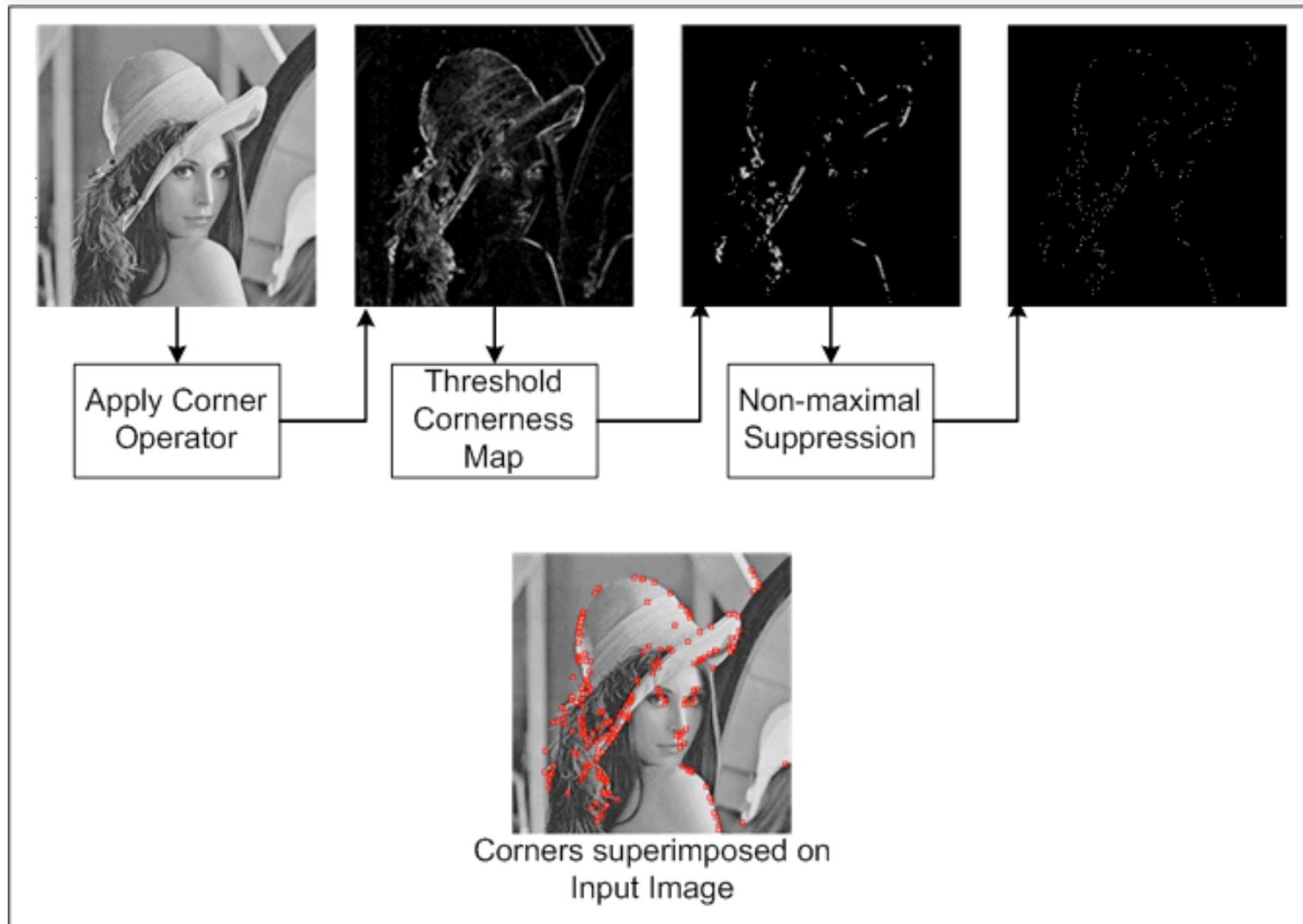


Algoritmos de detección de esquinas





Algoritmos de detección de esquinas





Algoritmos de detección de esquinas

○ Moravec (I)

○ Introducción

- Define el concepto de punto de interés
 - Regiones distintivas en una imagen
 - Útiles para encontrar correspondencias entre frames
- Qué son estos puntos?
 - Tienen gran variación de intensidad en todas direcciones
 - Es el caso de las esquinas, pero no sólo
 - No obstante, se admite esta definición

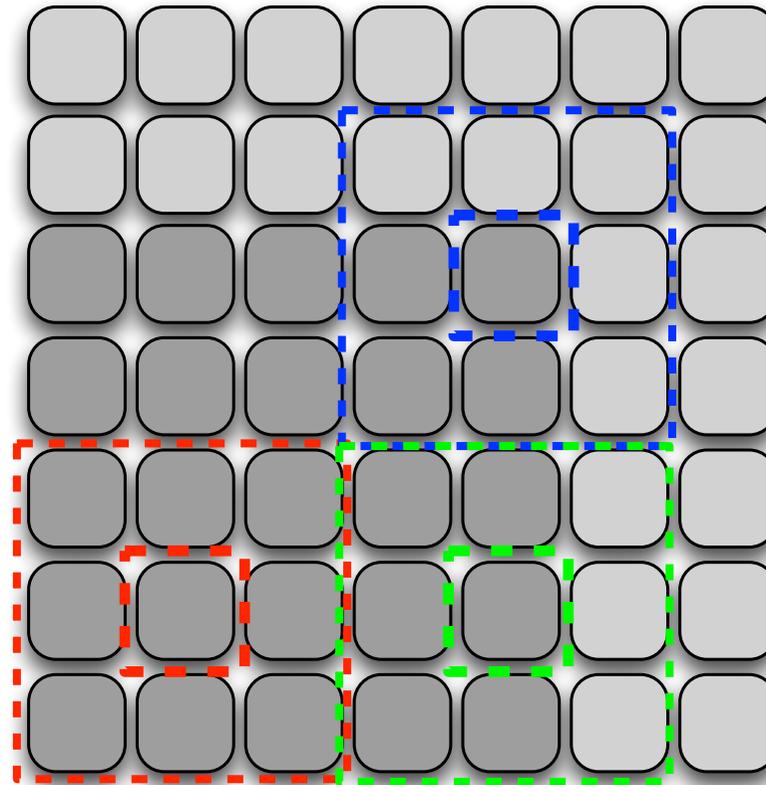


H. P. Moravec. [Towards Automatic Visual Obstacle Avoidance](#). *Proc. 5th International Joint Conference on Artificial Intelligence*, pp. 584, 1977.



Algoritmos de detección de esquinas

- Moravec (II)
 - Cómo funciona?





Algoritmos de detección de esquinas

○ Moravec (III)

○ Cómo se usa

- Se calcula el valor del operador para cada pixel
 - Mínimo de la diferencia entre los ocho desplazamientos
- En el conjunto de la imagen se crea un *cornerness map*

X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X	X	0	0	0	0	0	0	0	0	0	1	1	1	X	X
X	X	0	0	0	0	0	1	1	0	0	1	2	1	X	X
X	X	0	0	0	0	0	2	1	0	0	1	1	1	X	X
X	X	0	0	0	0	0	0	0	0	0	0	0	0	X	X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X





Algoritmos de detección de esquinas

- Moravec (IV)
 - Algunas observaciones
 - Una esquina determina un máximo local
 - Un punto aislado también provoca un máximo local
 - ⊙ Ambos tendrían el mismo valor
 - ⊙ Es un algoritmo sensible al ruido
 - ⊙ Aún aumentando la ventana, aparecería máximo local
 - En los límites de la imagen no se puede calcular (X)
 - En imágenes reales aparecen muchos máximos locales
 - ⊙ Hay que aplicar un umbral para eliminar los débiles





Algoritmos de detección de esquinas

○ Moravec (V)

○ Algoritmo

1. Para cada pixel (x,y), calcular variación ante cambio (u,v)

$$V(\Delta \mathbf{u}) = \sum_i w(x_i) [I_0(x_i + \Delta \mathbf{u}) - I_0(x_i)]^2$$

2. Construir el *cornerness map* con todos los pixeles (x,y)

$$C = \min (V(\Delta \mathbf{u}))$$

3. Umbralizar el *cornerness map*

$$C = \begin{cases} C & C \geq T \\ 0 & C < T \end{cases}$$

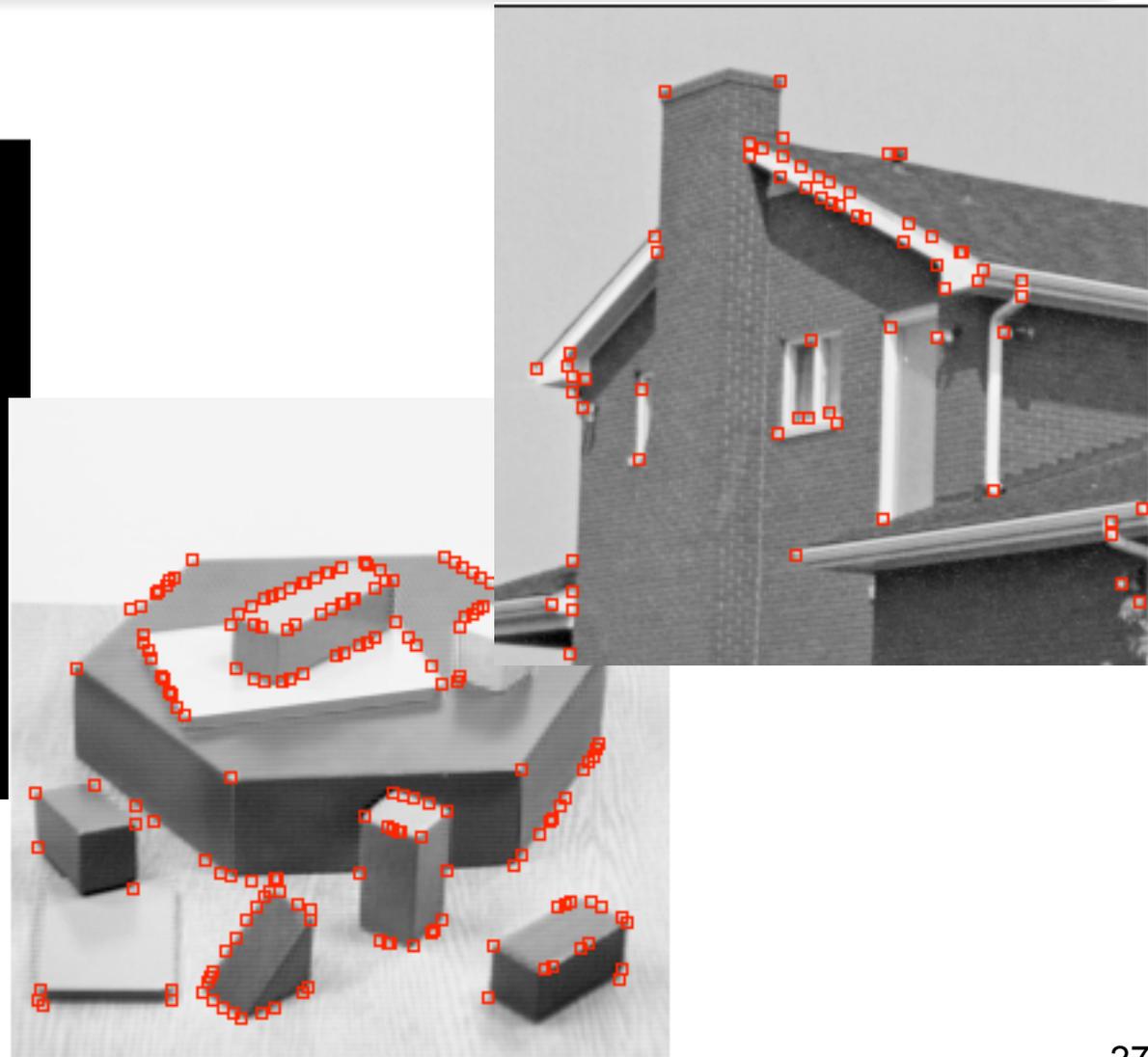
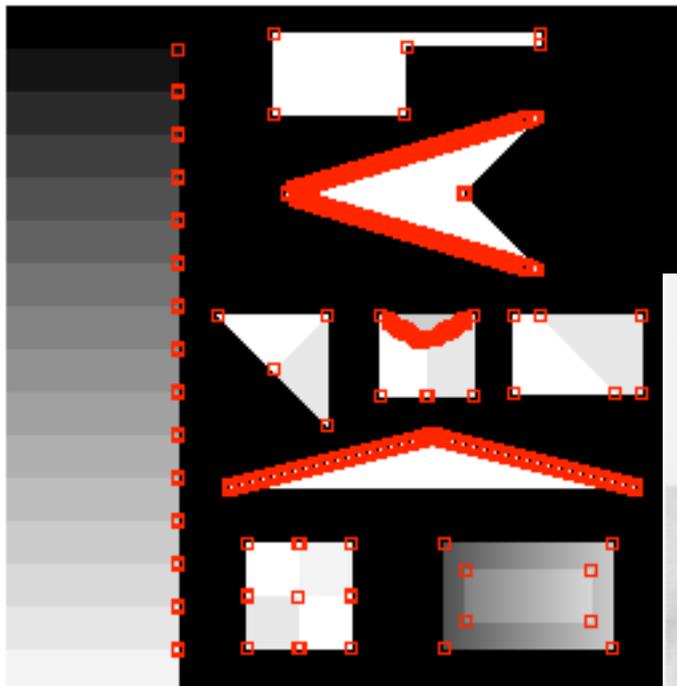
4. Suprimir los no máximos





Algoritmos de detección de esquinas

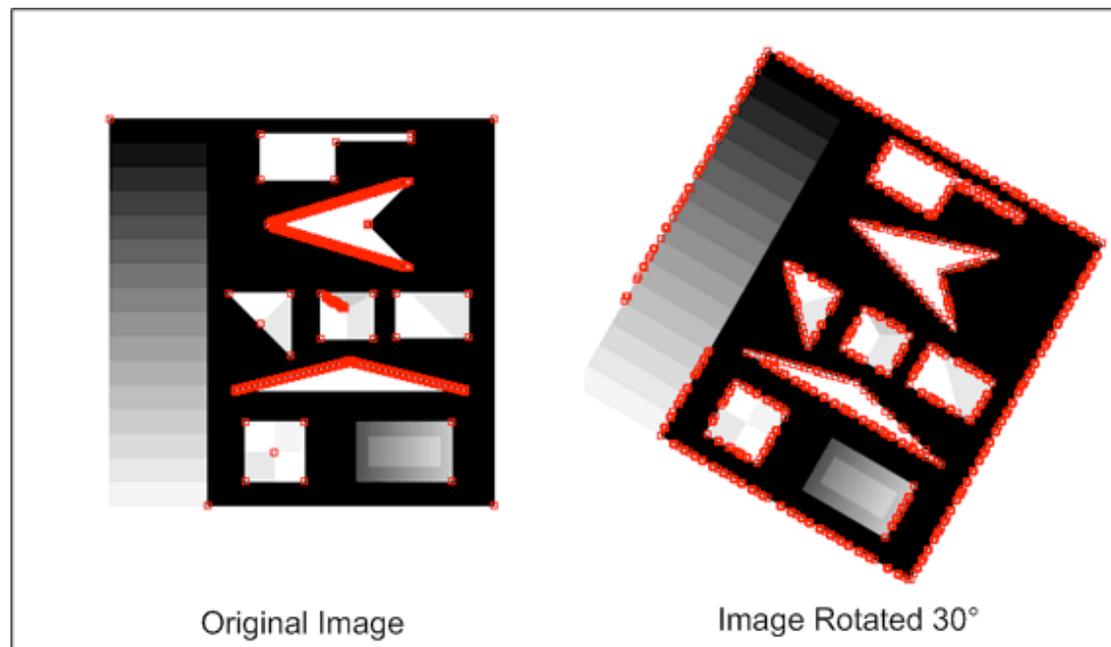
○ Moravec (VI)





Algoritmos de detección de esquinas

- Moravec (VII)
 - Respuesta anisotrópica
 - Varía respecto a giro si el ángulo no es múltiplo de 45
 - Poca repetibilidad del operador

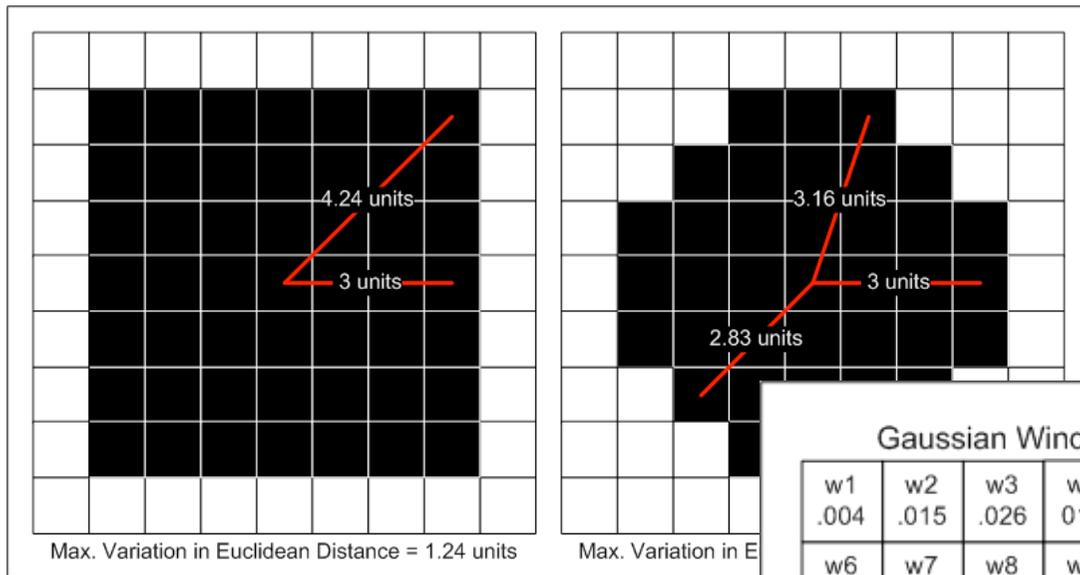




Algoritmos de detección de esquinas

○ Moravec (VII)

○ Ventana de análisis: cuadrada y binaria



Gaussian Window

w1	w2	w3	w4	w5
.004	.015	.026	.015	.004
w6	w7	w8	w9	w10
.015	.059	.095	.059	.015
w11	w12	w13	w14	w15
.026	.095	.15	.095	.026
w16	w17	w18	w19	w20
.015	.059	.095	.059	.015
w21	w22	w23	w24	w25
.004	.015	.026	.015	.004

Intensity Variation: $V = \sum_{i=1}^{25} w_i (A_i - B_i)^2$





Algoritmos de detección de esquinas

○ Moravec (IX)

○ Respuesta significativa en los contornos

X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X	X	0	0	0	0	0	0	0	0	0	1	1	1	X	X
X	X	1	2	1	0	0	1	1	0	0	1	2	1	X	X
X	X	2	2	2	0	0	2	1	0	0	1	1	1	X	X
X	X	1	1	1	0	0	0	0	0	0	0	0	0	X	X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X





Algoritmos de detección de esquinas

- Moravec (y X)
 - Conclusiones
 - Es sencillo, aunque se le considera obsoleto
 - No es invariante a rotación
 - Sensible al ruido: puntos aislados=esquinas
 - Muy eficiente computacionalmente





Algoritmos de detección de esquinas

○ Harris/Plessey (I)

○ Introducción

- Orientado al seguimiento de esquinas como de bordes
- Se centra en los problemas de Moravec
- Consigue mejor detección y repetibilidad
- Emplea más recursos computacionales
- Su uso es el más extendido
- Se le conoce como detector de Harris o de Plessey



C. Harris and M. Stephens. [A Combined Corner and Edge Detector](#). *Proc. Alvey Vision Conf.*, Univ. Manchester, pp. 147-151, 1988.



Algoritmos de detección de esquinas

○ Harris/Plessey (II)

- Difiere de Moravec en la medida de la autocorrelación
 - Permite su cálculo en cualquier dirección
 - Se basa en el desarrollo en serie de Taylor

$$\begin{aligned} V(\Delta \mathbf{u}) &= \sum_i w(x_i) [I_0(x_i + \Delta \mathbf{u}) - I_0(x_i)]^2 \\ &\approx \sum_i w(x_i) [I_0(x_i) + \nabla I_0(x_i) \cdot \Delta \mathbf{u} - I_0(x_i)]^2 \\ &= \sum_i w(x_i) [\nabla I_0(x_i) \cdot \Delta \mathbf{u}]^2 \\ &= \Delta \mathbf{u}^T \mathbf{A} \Delta \mathbf{u} \end{aligned}$$

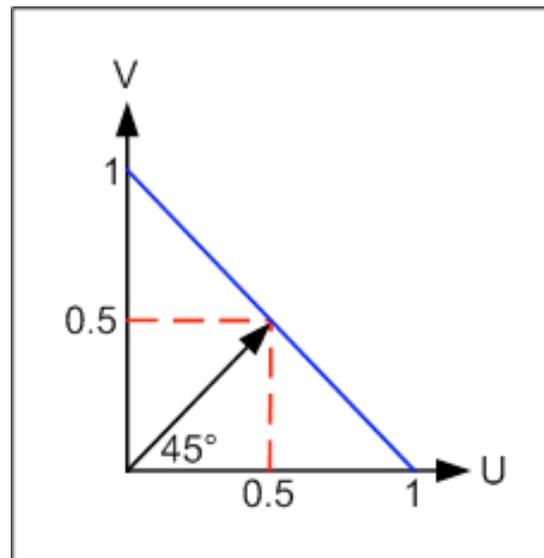
$$\nabla I_0(x_i) = \left(\frac{\partial I_0}{\partial x}, \frac{\partial I_0}{\partial y} \right) x_i \quad \mathbf{A} = w * \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$





Algoritmos de detección de esquinas

- Harris/Plessey (III)
 - El resultado es distinto al de Moravec
 - Se basa en el uso del tensor estructural: **A**
 - El operador sigue siendo anisotrópico
 - Se basa en los gradientes en horizontal y vertical
 - Permite variación de intensidad en cualquier dirección





Algoritmos de detección de esquinas

- Harris/Plessey (IV)
 - Respuesta en los contornos
 - Mejora a Moravec midiendo en diferentes direcciones
 - ⊙ Moravec sólo maneja las ocho vecinas
 - La matriz **A**...
 - ⊙ ... contiene información de la geometría de la imagen
 - Sus autovalores...
 - ⊙ ... son proporcionales a sus curvaturas principales
 - ⊙ ... y por lo tanto invariantes a rotación de **A**
 - Sin embargo...
 - ⊙ ... no son invariantes a rotación de la imagen
 - ⊙ ... porque los gradientes son horizontal y vertical

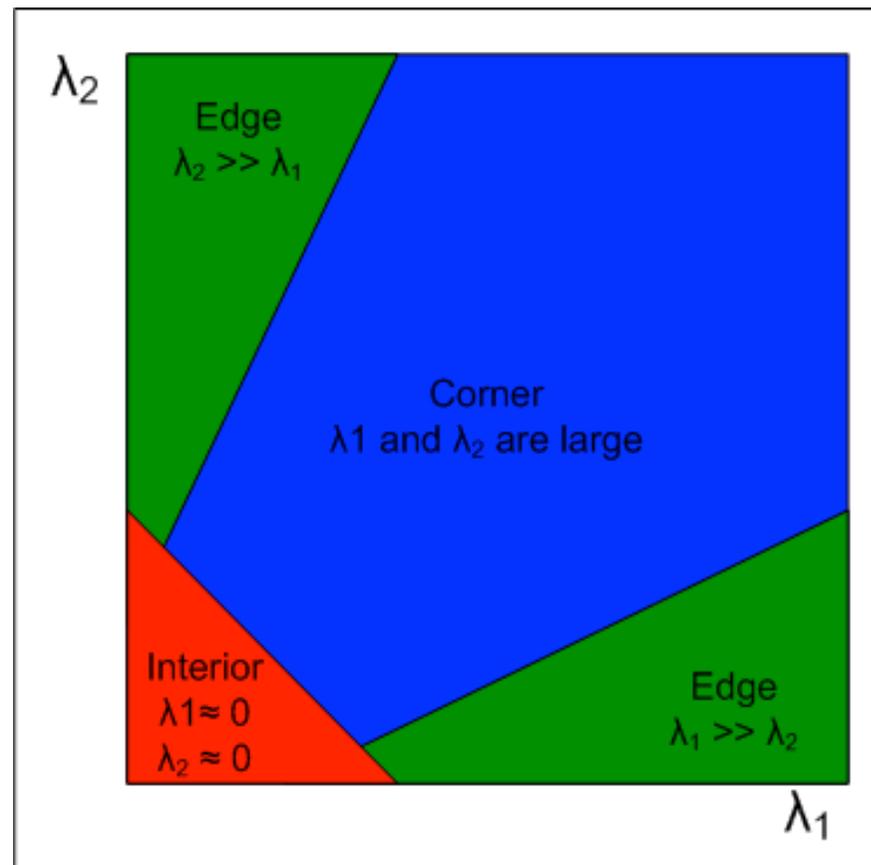




Algoritmos de detección de esquinas

○ Harris/Plessey (V)

○ Sean λ_1, λ_2 los autovalores de **A**



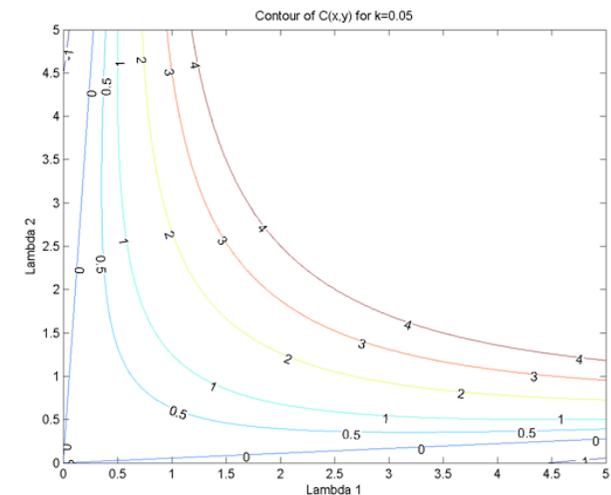
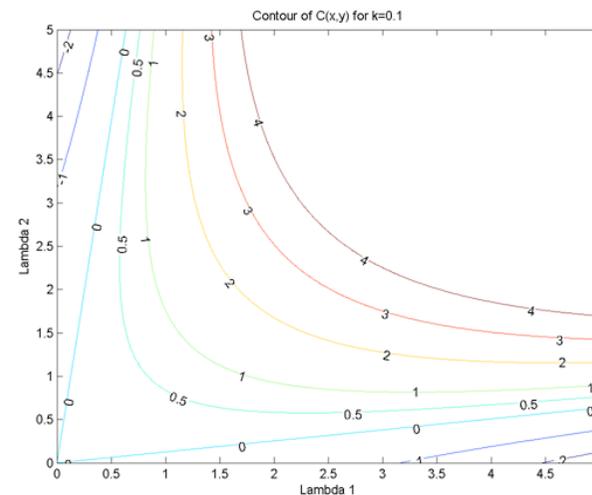
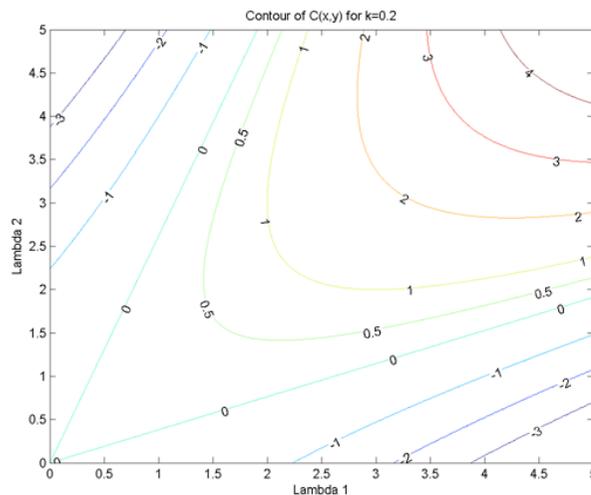


Algoritmos de detección de esquinas

○ Harris/Plessey (VI)

- Sobre esta información se calcula el *cornerness*

$$C = \det(\mathbf{A}) - k \text{traza}(\mathbf{A})^2 = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2$$





Algoritmos de detección de esquinas

○ Harris/Plessey (VII)

○ Algoritmo

1. Calcular la matriz **A**
2. Construir el *cornerness map*, para cada pixel

$$C = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2$$

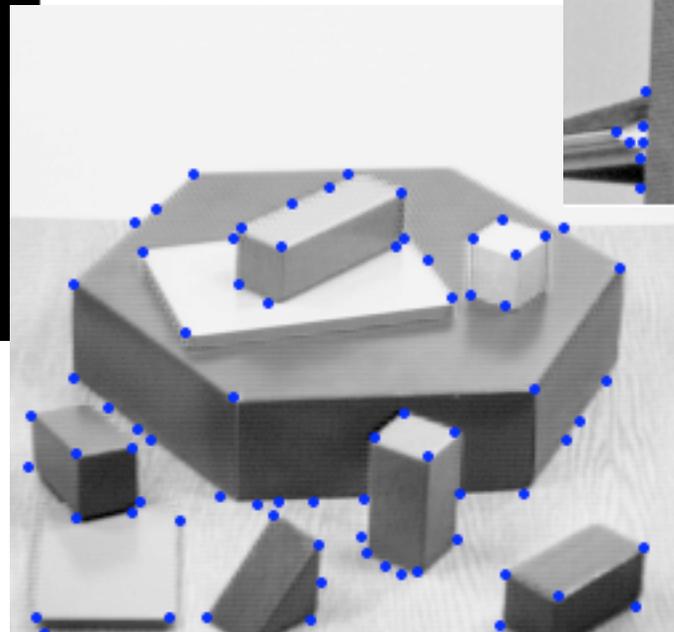
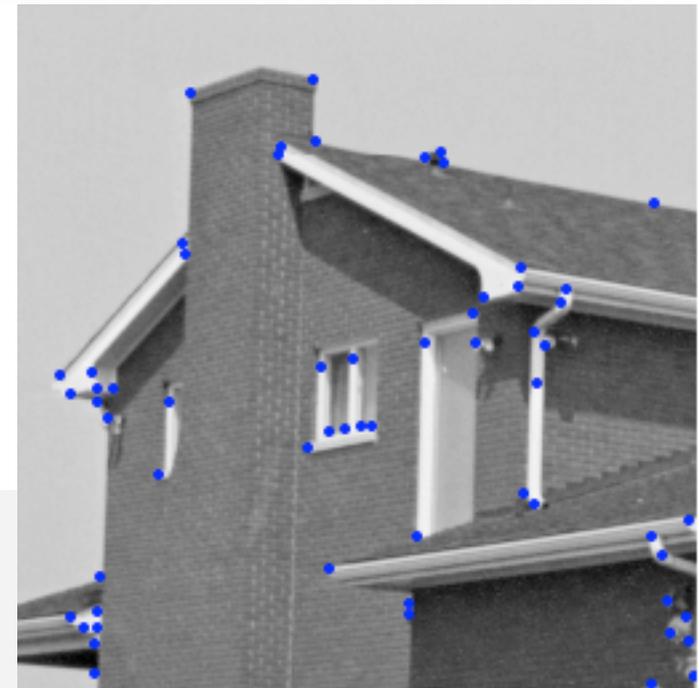
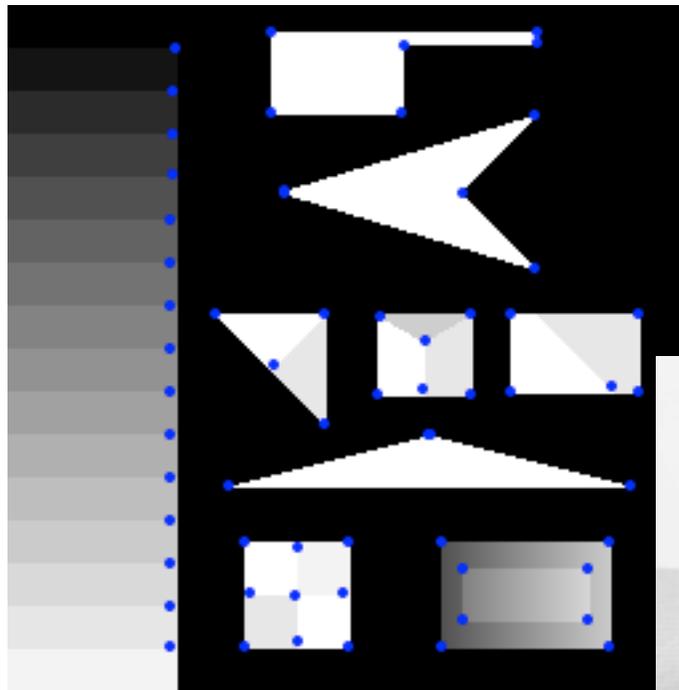
3. Umbralizar el mapa, conservando los valores altos
4. Suprimir los no máximos





Algoritmos de detección de esquinas

○ Harris/Plessey (VIII)

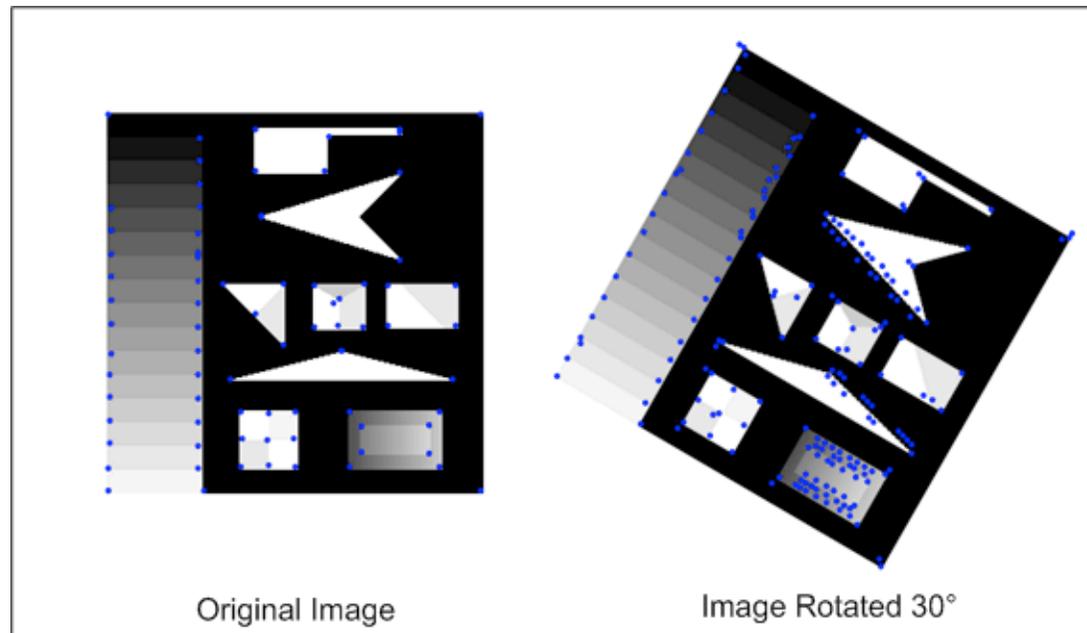




Algoritmos de detección de esquinas

○ Harris/Plessey (IX)

- Es computacionalmente costoso, requiere convolución
- Sensibilidad al ruido, por el uso del gradiente
- Buena respuesta sólo en esquinas en L
- Respuesta anisotrópica, por los gradientes direccionales





Algoritmos de detección de esquinas

- Harris/Plessey (y X)
 - Conclusiones
 - Se usa ampliamente por detección y repetibilidad
 - Tiene algunos inconvenientes
 - Sensibilidad al ruido: ampliar la ventana de cálculo
 - ◎ Más coste computacional
 - ◎ Peor detección
 - Uso de gradientes direccionales
 - ◎ Aún así, mejora los resultados de Moravec





Tabla de Contenidos

- Introducción
- Algoritmos de detección de esquinas
- SIFT



SIFT

- Introducción
 - Detección de esquinas para imágenes similares
 - Misma escala
 - Misma orientación, ...
 - SIFT (*Scale Invariant Feature Transform*) para distinta...
 - ... escala
 - ... rotación
 - ... iluminación
 - ... punto de vista

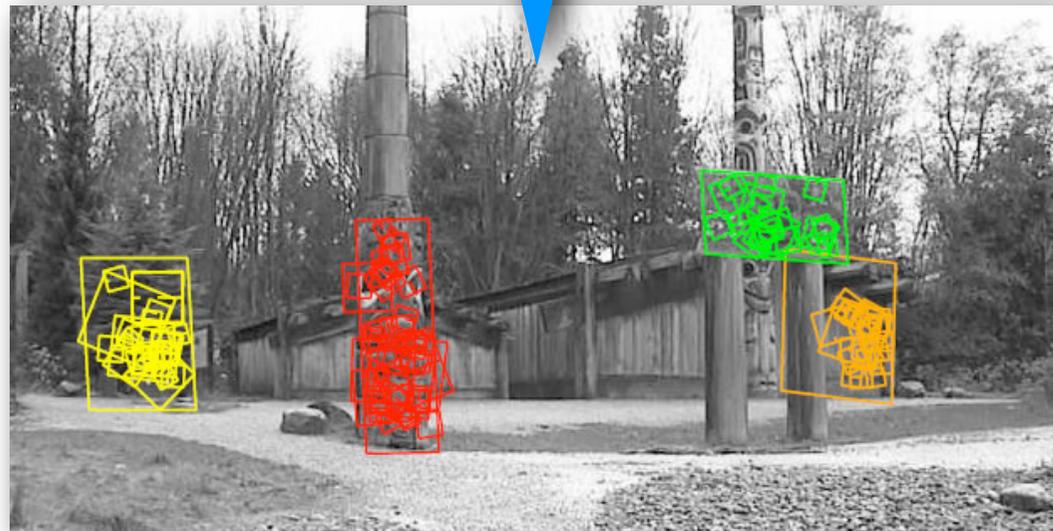


David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60, 2 (2004), pp. 91-110

Autor: S. Domínguez



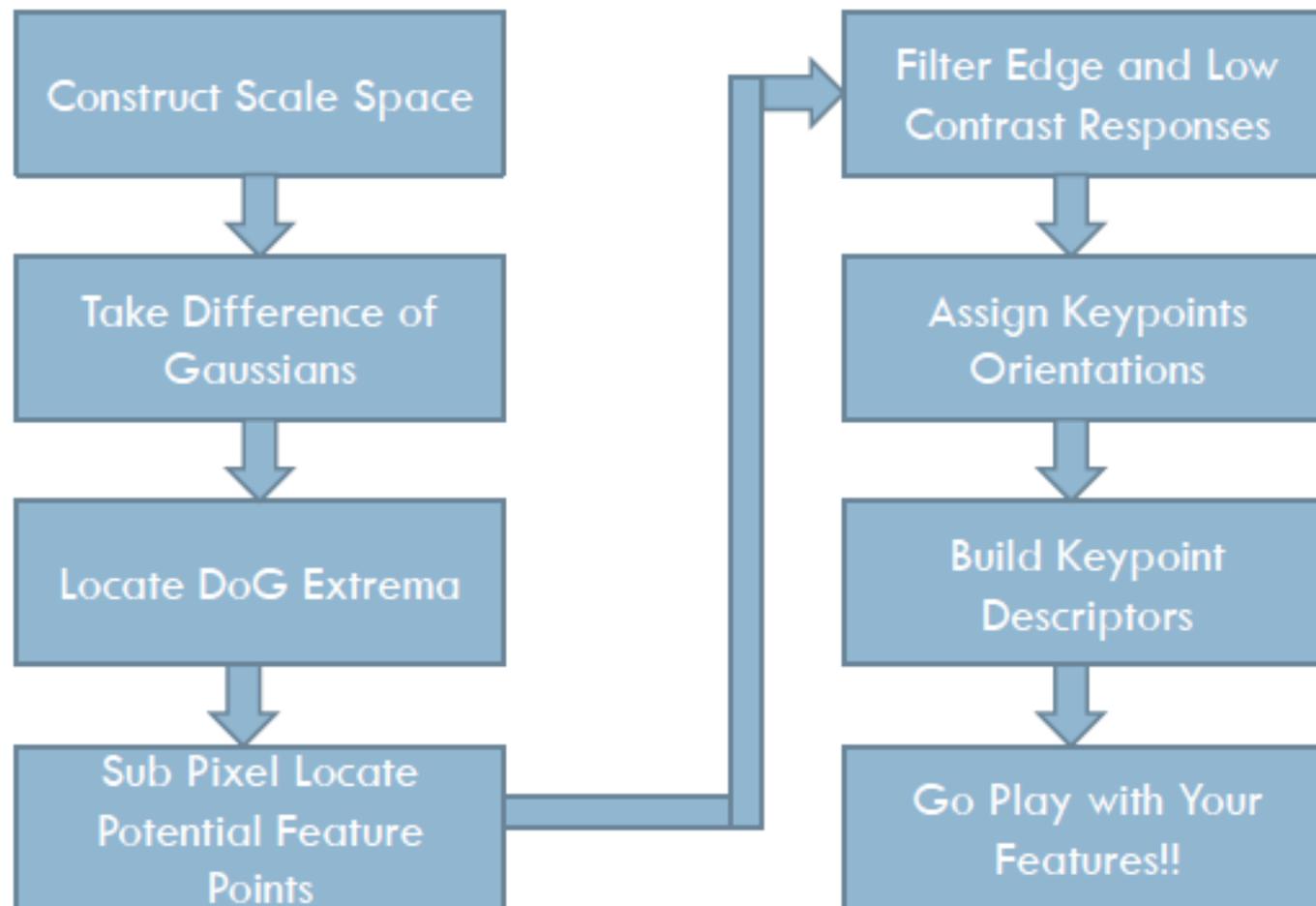
SIFT





SIFT

○ El algoritmo





SIFT

- Construir el espacio de escala
 - El mismo objeto observado a distintas escalas:
 - Una hoja del árbol: especie, presencia de plagas, ...
 - El árbol completo: edad conformación, poda, ...
 - Eliminar detalles irrelevantes a cada escala:
 - La única forma es un filtrado gaussiano (*)
 - Para generar el espacio de escala:
 - Tomar la imagen de origen
 - Generar copias difuminadas progresivamente



(*) Lindeberg (1994) [Scale-Space Theory in Computer Vision](#), Kluwer Academic Publishers/Springer, Dordrecht, Netherlands, 1994.

Autor: S. Domínguez



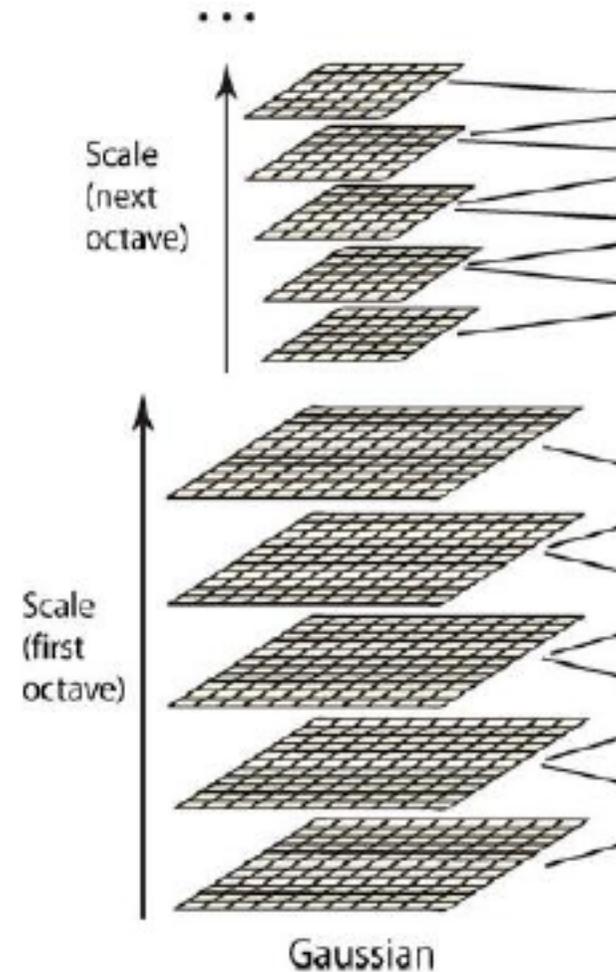
SIFT





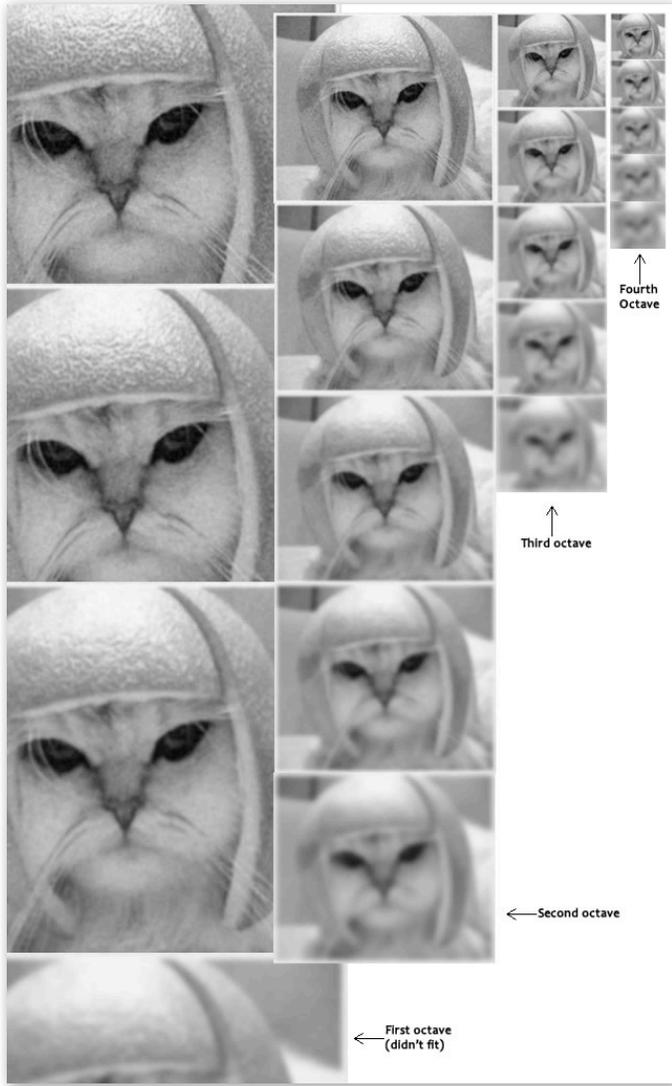
SIFT

- Espacio de escalas en SIFT
 - Primero se generan copias difuminadas del original
 - Luego se reescala el original a la mitad de tamaño
 - Se generan copias difuminadas de la reducción
 - Se continúa con el esquema
 - Escalado
 - Difuminación





SIFT





SIFT

- Los detalles (I)
 - Octavas y escalas
 - Su número es decisión del programador
 - Lowe sugiere 4 octavas y cinco escalas
 - Primera octava
 - Se puede generar escalando la imagen al doble
 - Luego se filtra para evitar aliasing
 - Así se generan cuatro veces más características
 - Más información para establecer correspondencias





SIFT

- Los detalles (II)
 - Difuminado
 - Mediante convolución con gaussiana

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$$

- L es la imagen difuminada
- G es la máscara gaussiana
- I es la imagen
- (x,y) las coordenadas del pixel
- * es el operador de convolución





SIFT

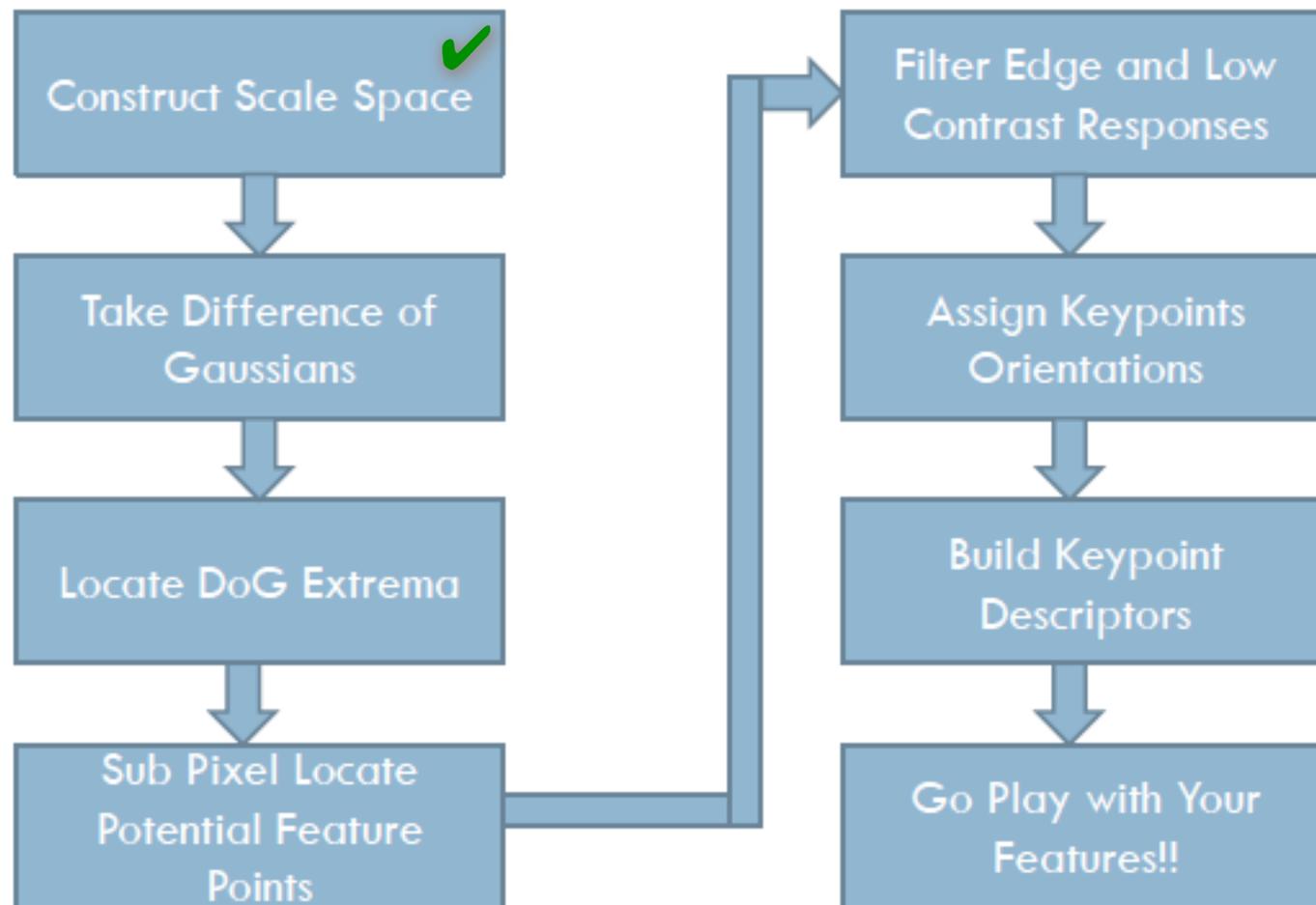
- Los detalles (y III)
 - Cantidad de difuminado
 - Con un factor de $\sqrt{2}$

	scale →				
octave	0.707107	1.000000	1.414214	2.000000	2.828427
	1.414214	2.000000	2.828427	4.000000	5.656854
	2.828427	4.000000	5.656854	8.000000	11.313708
	5.656854	8.000000	11.313708	16.000000	22.627417



SIFT

○ El algoritmo





SIFT

- LoG: *Laplacian of Gaussian*
 - Laplaciana de gaussiana
 - Se toma una imagen
 - Se difumina con un filtro gaussiano
 - Se calcula su segunda derivada: laplaciana
 - Detecta bordes y esquinas aptos como keypoints
 - Por qué no sólo laplaciana?
 - Es muy sensible al ruido
 - El filtrado gaussiano lo disminuye y mejora el resultado
 - Problema
 - Calcular la laplaciana es computacionalmente costoso





SIFT

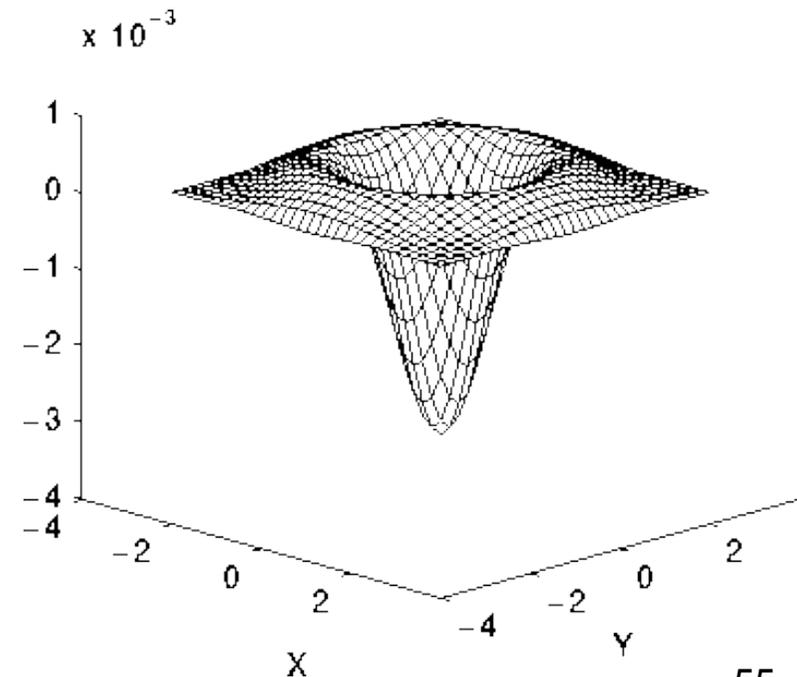
○ Laplaciana de gaussiana

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$$

$$\nabla^2 L(x, y) = \frac{\partial^2 L}{\partial x^2} + \frac{\partial^2 L}{\partial y^2} = L_{xx} + L_{yy}$$

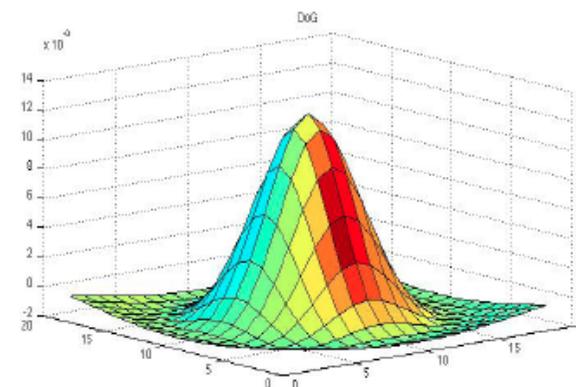
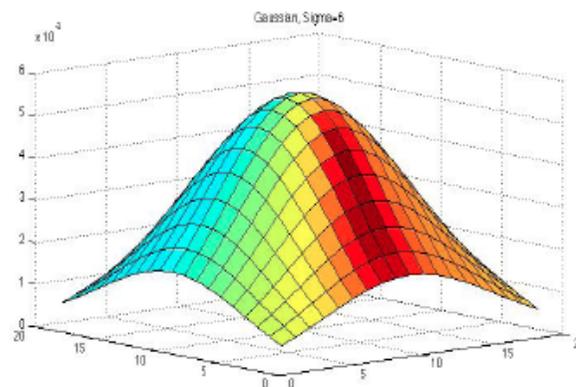
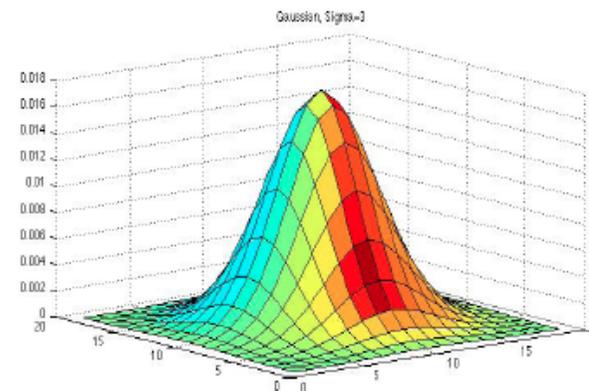
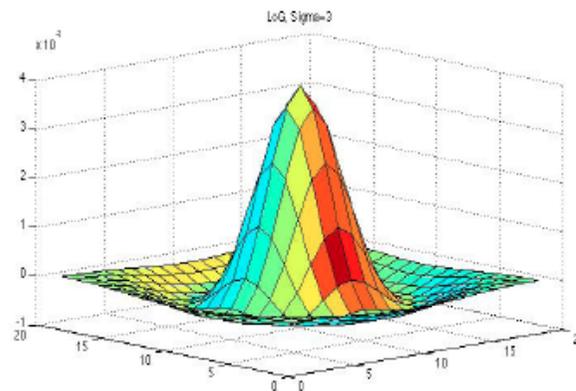
$$\nabla^2 L(x, y) = -\frac{1}{\pi\sigma^4} \left[1 - \frac{x^2 + y^2}{2\sigma^2} \right] e^{-\frac{x^2+y^2}{2\sigma^2}}$$





SIFT

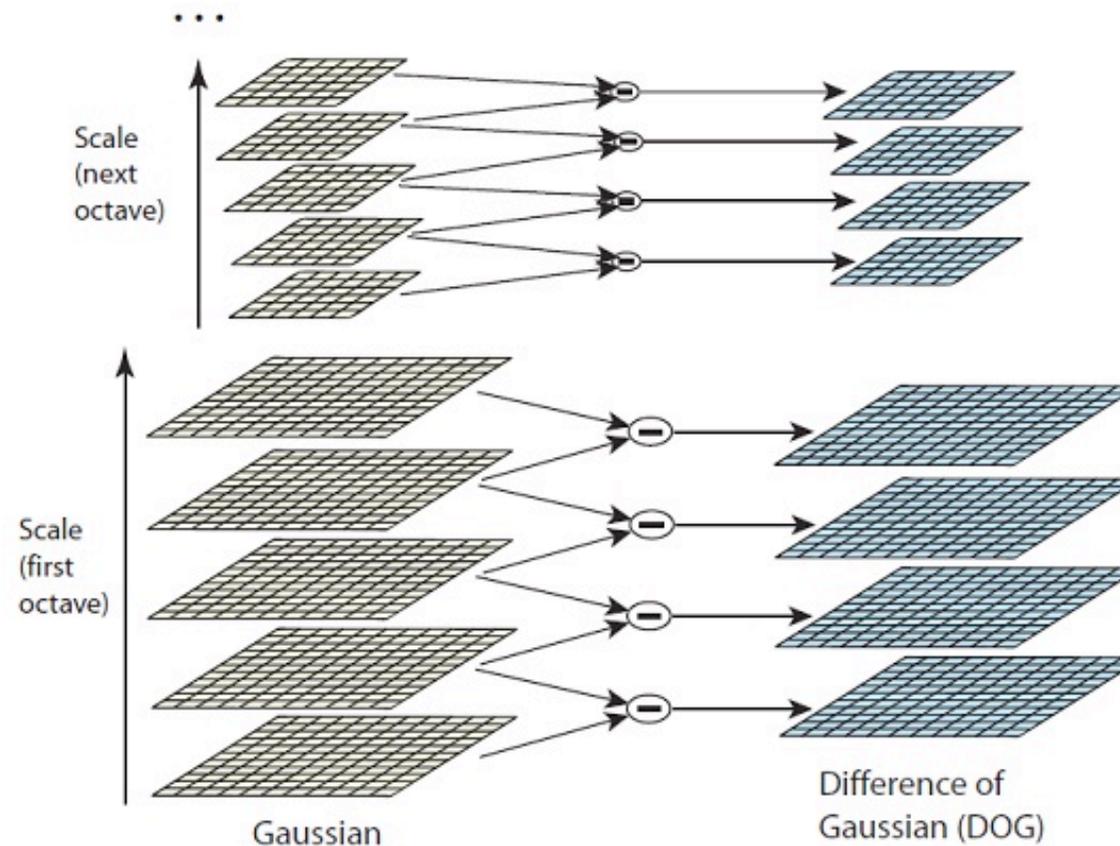
- Aproximación de LoG: DoG (*Difference of Gaussians*)
- Permite generar una aproximación rápidamente





SIFT

- Procedimiento de aproximación
 - Hallar la diferencia entre dos escalas consecutivas





SIFT

○ Ventajas de DoG

● LoG no es invariante a escala:

- Debido a la expresión de la Gaussiana

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$$

- Entonces LoG tampoco lo es
- Hay que conseguir invarianza a escala

$$\nabla^2 G \longrightarrow \sigma^2 \nabla G$$

● DoG es invariante a escala

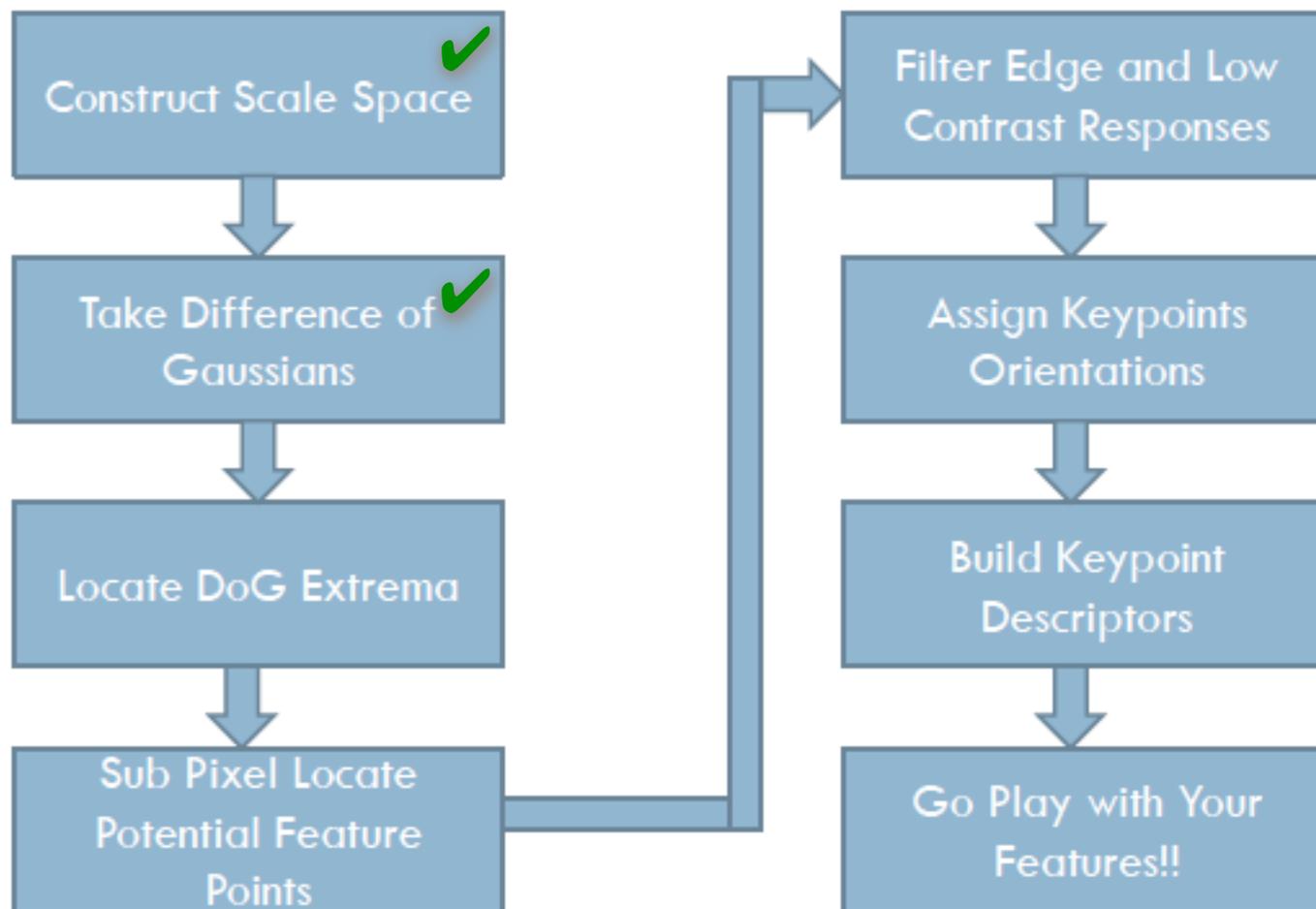
- El resultado ya está multiplicado





SIFT

○ El algoritmo





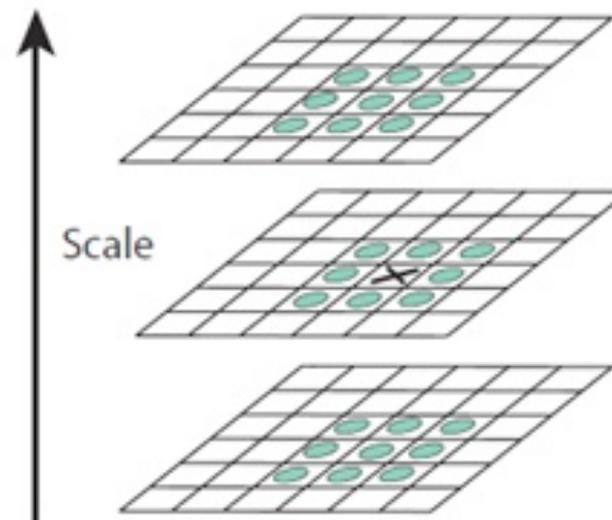
SIFT

- Determinar keypoints
 - La determinación de keypoints consta de dos fases
 1. Localizar extremos en las imágenes DoG
 2. Determinar su posición con precisión subpixel



SIFT

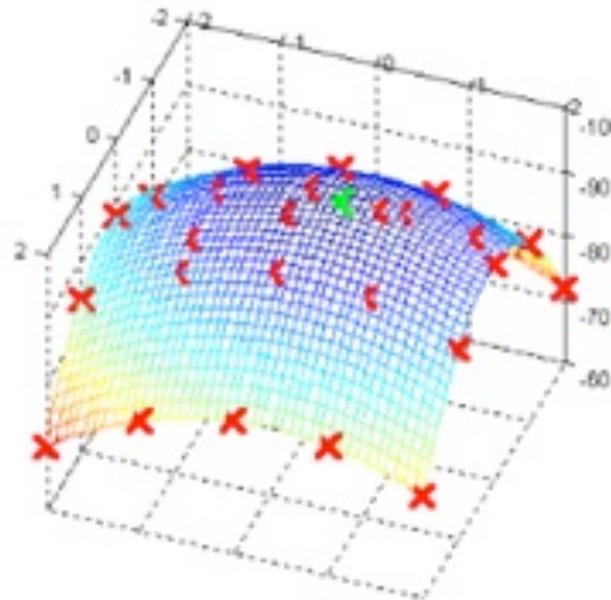
- Determinar keypoints
 - Localizar extremos en las imágenes DoG
 - Se toma un pixel y se consideran sus 26 vecinos
 - En la misma escala: 8
 - En la escala superior e inferior: 9 para cada una
 - Si es el máximo/mínimo de todos ellos se selecciona





SIFT

- Determinar keypoints
 - Determina su posición con precisión subpixel (i)
 - Los candidatos no coinciden exactamente con el extremo
 - Este no suele coincidir con la posición exacta del pixel
 - Hay que calcular su posición con algoritmos subpixel





SIFT

- Determinar keypoints
 - Determinar su posición con precisión subpixel (ii)
 - Se realiza el desarrollo en serie alrededor del candidato

$$D(\mathbf{x}) = D + \frac{\partial D}{\partial \mathbf{x}} \mathbf{x} + \frac{1}{2} \mathbf{x}^T \frac{\partial^2 D}{\partial \mathbf{x}^2} \mathbf{x}$$

- Derivando e igualando a cero se obtiene la posición
 - Ahora con precisión subpixel

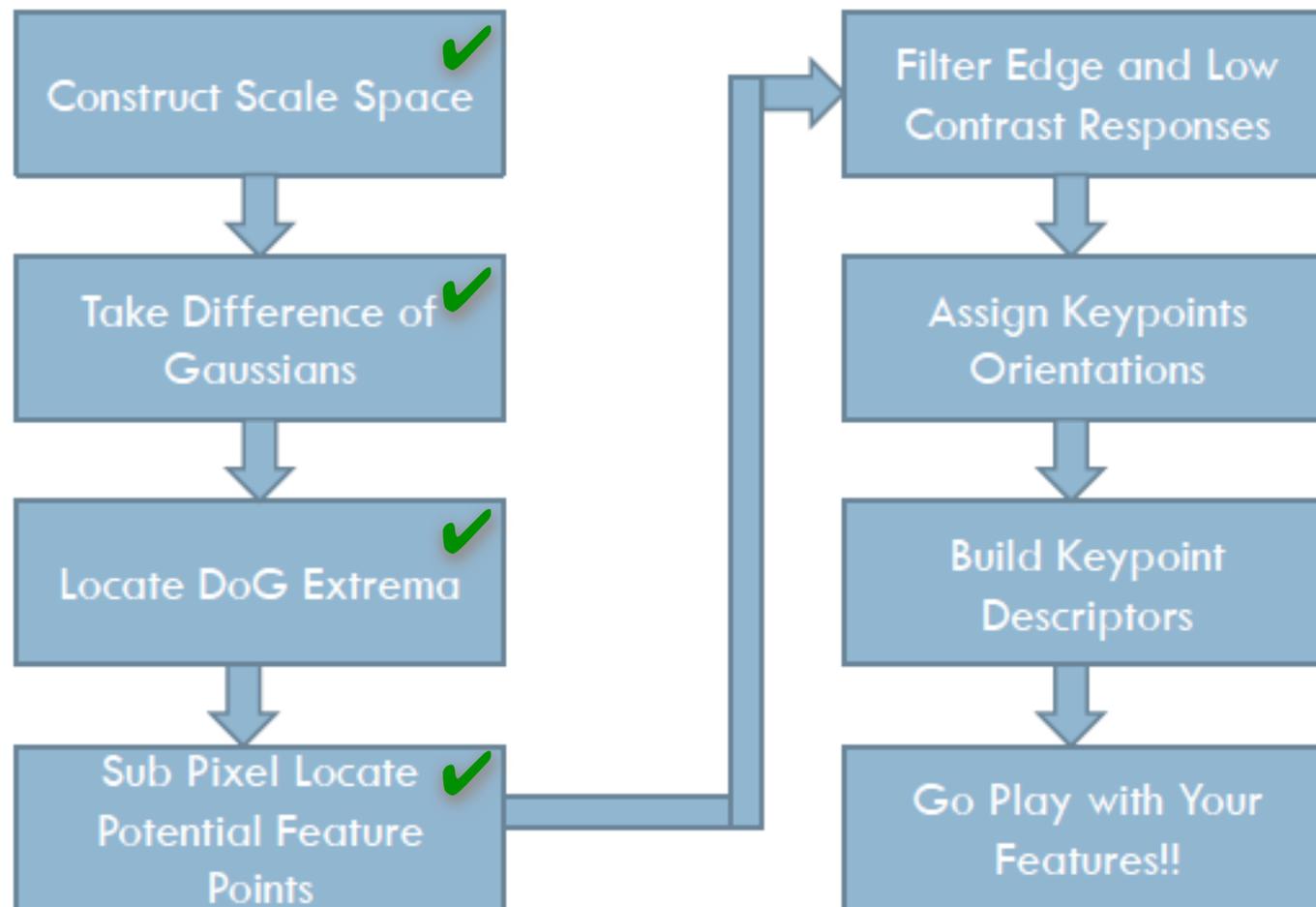
$$\hat{\mathbf{x}} = -\frac{\partial^2 D}{\partial \mathbf{x}^2}^{-1} \frac{\partial D}{\partial \mathbf{x}}$$





SIFT

○ El algoritmo





SIFT

- Eliminación de keypoints de bajo contraste
 - Se evalúa la magnitud de DoG de cada candidato
 - No se tiene en cuenta el signo
 - Dado que la posición es subpixel, se debe estimar
 - Usando también el desarrollo en serie
 - Si la magnitud no supera un umbral se desecha





SIFT

- Eliminación de contornos (I)
 - Para cada keypoint se calculan dos gradientes
 - Perpendiculares entre si
 - Existen tres posibles combinaciones
 - Región plana
 - ⊙ Ambos gradientes son pequeños
 - Un contorno
 - ⊙ Un gradiente es alto (perpendicular al contorno)
 - ⊙ El otro es bajo (a lo largo del contorno)
 - Una esquina
 - ⊙ Ambos gradientes son altos
 - Las esquinas son los mejores keypoints
 - Se rechaza lo que no sean esquinas



SIFT

- Eliminación de contornos (y II)
 - Determinación de esquinas
 - Muy similar al criterio para el detector de Harris/Plessey
 - Se calcula la matriz Hessiana de la imagen DoG
 - Se conservan los keypoints para los que se cumple

$$\frac{\text{traza}(H)^2}{\det(H)} < \frac{r + 1}{r}$$

$$\text{traza}(H) = D_{xx} + D_{yy}$$

$$\det(H) = D_{xx}D_{yy} - D_{xy}^2$$

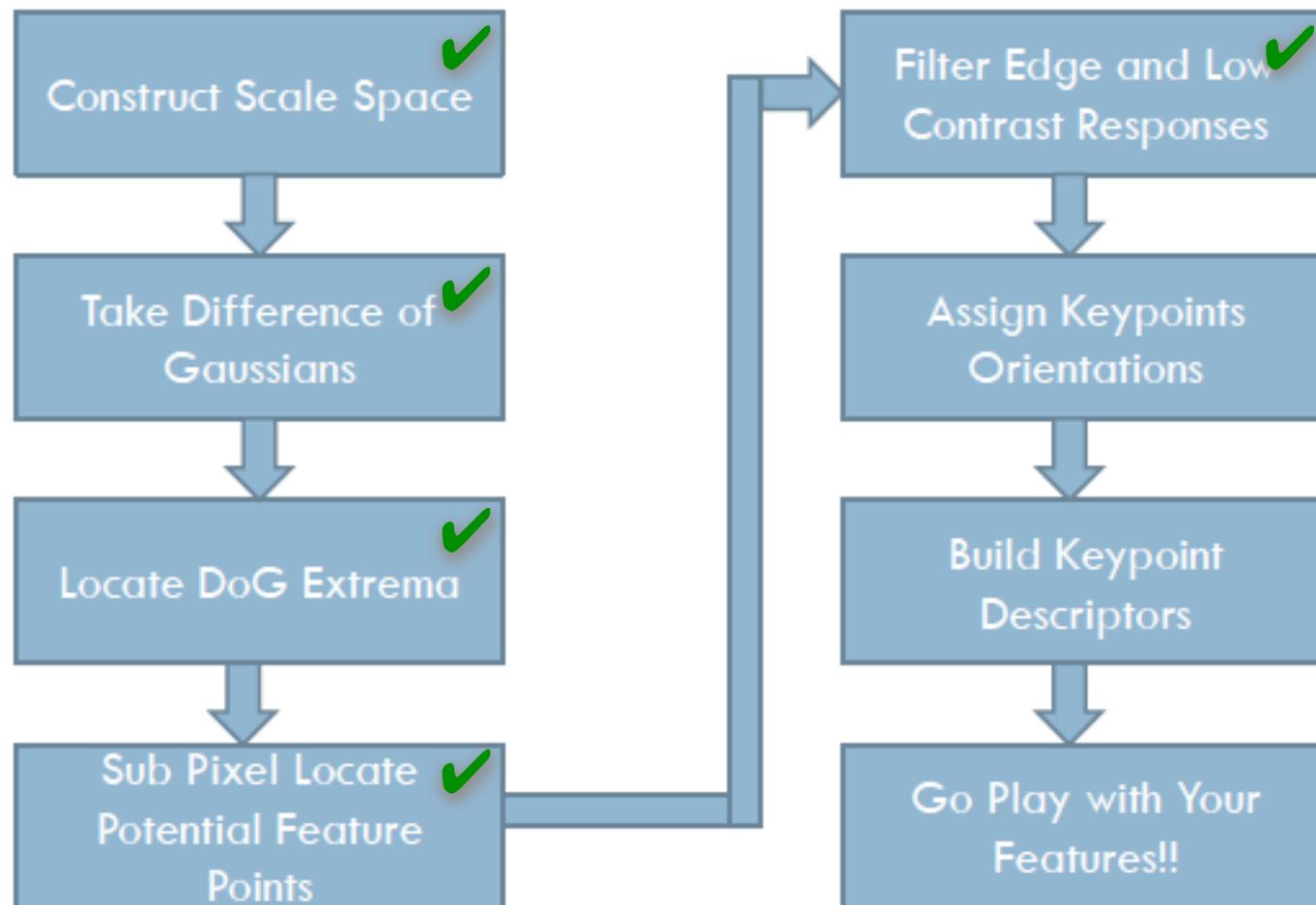
- Se suele usar $r=10$





SIFT

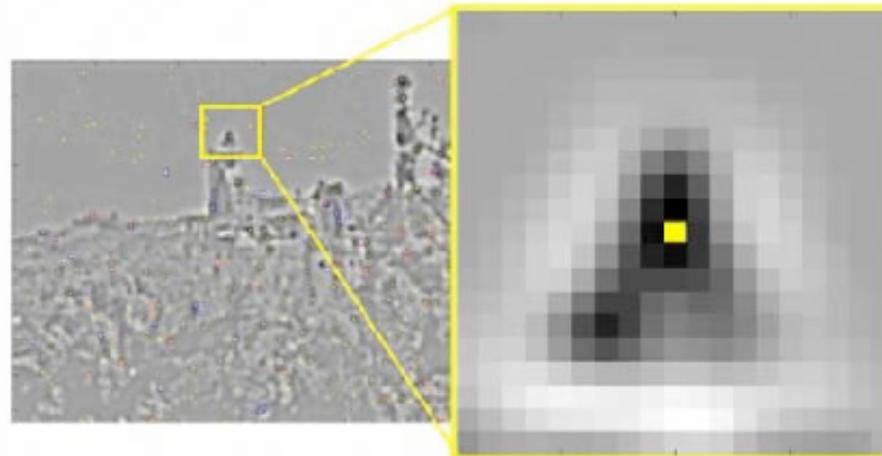
○ El algoritmo





SIFT

- Asignar orientación a los keypoints
 - Se toma una región alrededor del keypoint
 - 1.5 veces mayor que la ventana de la escala del keypoint
 - Se recopila la información del gradiente en la región
 - Magnitudes y orientaciones
 - Se toman las orientaciones predominantes
 - Se asignan an keypoint



A keypoint



SIFT

○ Procedimiento

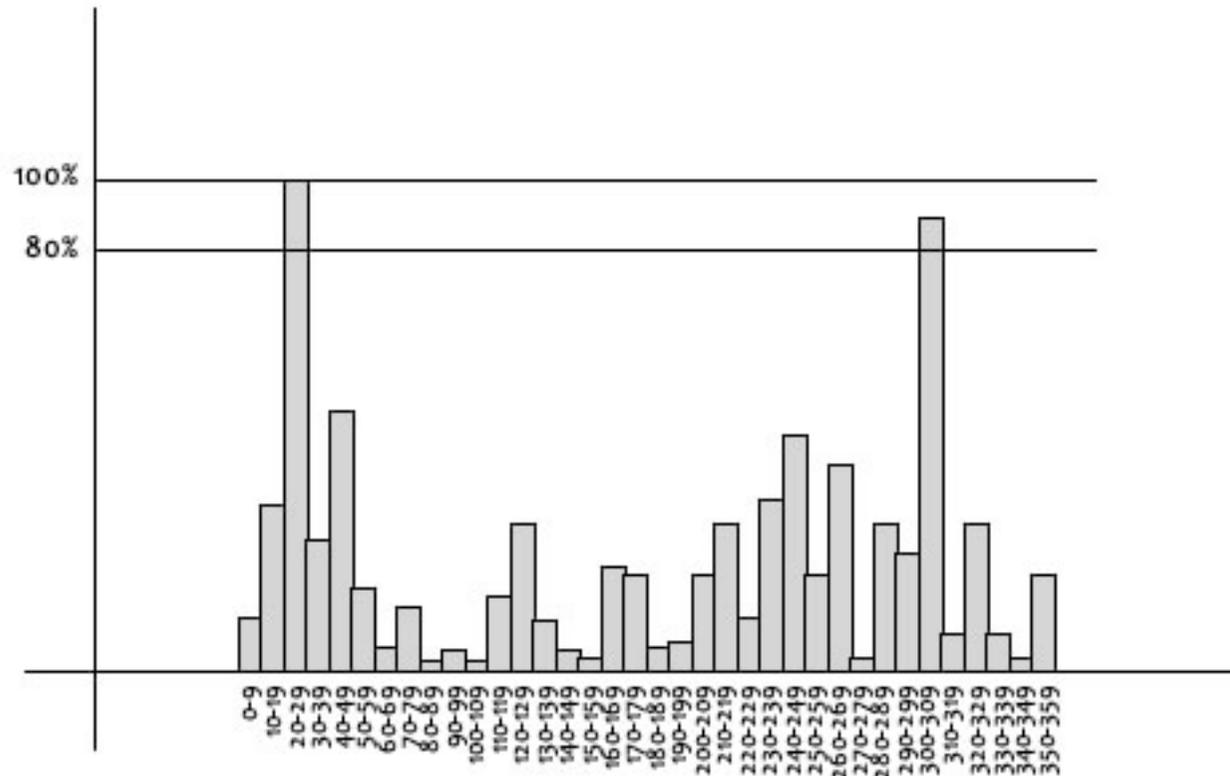
- Se calculan los gradientes en cada punto de la ventana
- Se construye un histograma de 36 bins
 - Cada uno, un tramo de 10 grados en orientación
 - Se pondera la aportación de cada pixel
 - Con su magnitud
 - Con el coeficiente de un filtro gaussiano de 1.5 escala
 - Se toma el pico máximo del histograma
 - Un keypoint por cada pico mayor que el 80% del máximo
 - Misma posición, distinta orientación
 - Se ajusta la orientación de cada uno
 - Ajuste de una parábola al pico y dos adyacentes





SIFT

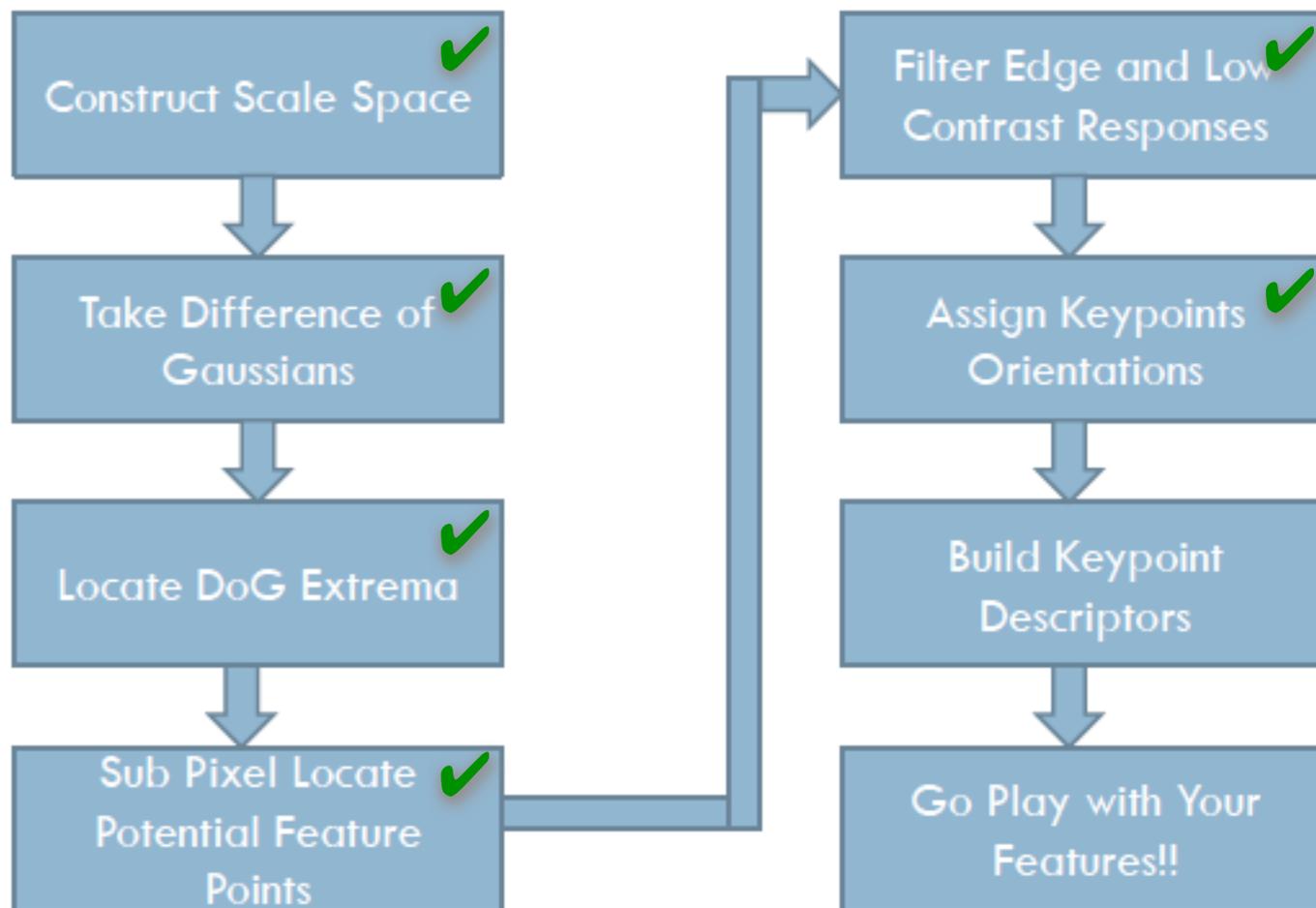
○ Procedimiento





SIFT

○ El algoritmo





SIFT

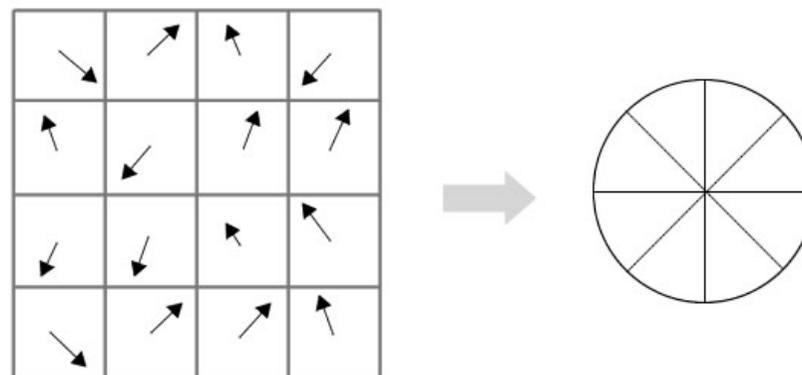
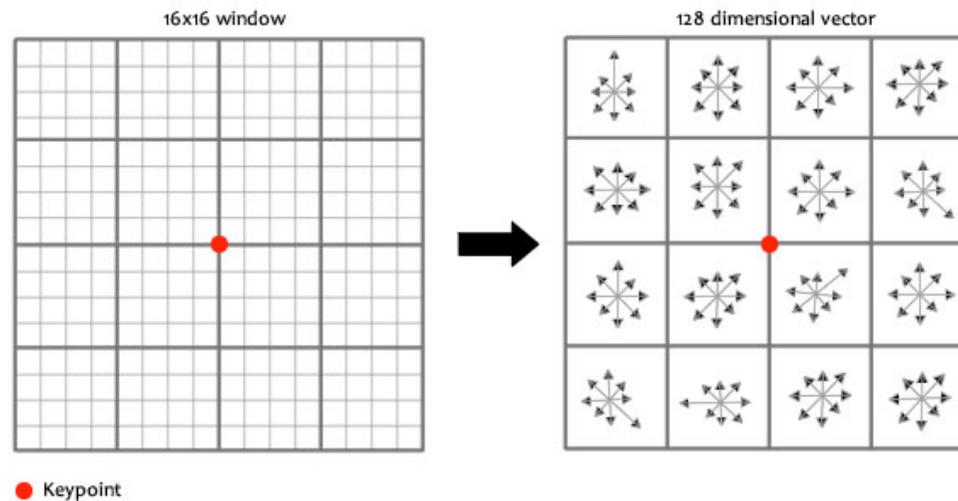
- Generación del descriptor
 - Se toma una ventana de 16x16 alrededor del keypoint
 - Se divide en 16 ventanas de 4x4
 - Se calculan los gradientes en cada una
 - Se pasan los gradientes a un histograma
 - Con ponderación gaussiana de 0.5 la escala
 - El histograma es de 8 bins en orientación
 - Con esto se genera el descriptor
 - $4 \times 4 \times 8 = 128$ números por cada keypoint
 - Finalmente, se normaliza este vector





SIFT

○ Generación del descriptor





SIFT

○ Problemas

● Invarianza a rotación

- Si se gira la imagen giran los gradientes

- El descriptor cambia

● Solución

- Restar orientaciones del descriptor y del keypoint

- Se consigue invarianza a rotación

● Independencia a la iluminación

- La saturación afecta a la magnitud del gradiente

- No afecta a la rotación

● Solución

- Las componentes del descriptor mayores que 0.2 se truncan

- El vector resultante se vuelve a normalizar





SIFT

○ Prestaciones

○ Robustez

- Repetibilidad del 80% ante...
 - ... 10% de ruido en la imagen
 - ... rotación de 45 grados
 - ... en bases de datos entre 1K y 100K keypoints
- Evaluado como mejor descriptor (*)



(*) K. Mikolajczyk, C. Schmid (2005) [A performance evaluation on local descriptors](#), IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, N. 10, pp. 1615-1630, octubre 2005

Autor: S. Domínguez